Introduzione

A due anni dal suo concepimento, Java rappresenta la tecnologia dominante per portare il contenuto interattivo nel World Wide Web. In un modo dove tutti hanno la propria opinione sul futuro del Web, non è cosa da poco.

Che cos'è Java?

Subito al punto: Java è un linguaggio di programmazione orientato agli oggetti sviluppato dalla Sun Microsystems, particolarmente adatto all'uso su Internet.

La *programmazione orientata agli oggetti* è un modo insolito ma potente di sviluppare software: un programma è considerato come un gruppo di oggetti che interagiscono tra loro.

La caratteristica più nota di Java è il fatto che può essere utilizzato per creare programmi che vengono eseguiti dalle pagine del World Wide Web. Questi programmi si chiamano *applet*.

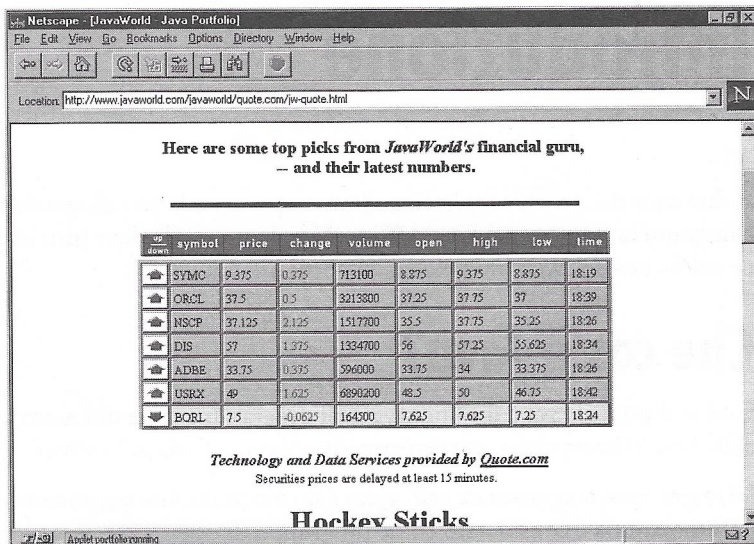
I programmi Java hanno riscosso grande successo nel Web perché hanno introdotto l'interattività in un mezzo che prima era quasi sempre a senso unico. Il Web distribuisce quasi tutte le informazioni in modo passivo: un utente con un browser richiede una pagina, la legge, ne chiede un'altra, la legge e così via.

Un applet Java in esecuzione su una pagina Web offre un'esperienza molto più ricca, sia in termini di informazioni sia per l'interazione dell'utente. Le informazioni possono cambiare in risposta all'input dell'utente, o essere aggiornate dinamicamente durante la visualizzazione della pagina Web.

Nella Figura 1 è illustrato un esempio di applet Java che aggiorna se stesso in modo dinamico. Questo applet, offerto dalla rivista Java World (`http://www.javaworld.com`) insieme con Quote.com, aggiorna un portafoglio azionario con le quotazioni in tempo reale.

Anche se i programmi basati sul Web rappresentano il suo punto di forza, Java è un linguaggio di programmazione di carattere generale che può essere utilizzato per sviluppare qualsiasi tipo di programma.

Un programma Java è creato come file di testo con estensione `.java` e compilato in uno o più file di bytecode con estensione `.class`. I *bytecode* sono serie di istruzioni simili a quelle in codice macchina create dalla compilazione di un programma; la differenza è che il codice macchina deve essere eseguito sul computer per il quale è stato compilato, mentre i bytecode possono essere eseguiti su qualsiasi sistema in grado di gestire i programmi Java.

Figura 1
Un applet Java che aggiorna un portafoglio azionario in tempo reale (per gentile concessione della rivista JavaWorld).

I motivi del successo internazionale di Java

In breve tempo Java è divenuto uno dei linguaggi di programmazione più diffuso, dentro e fuori Internet. C'è stato un vero e proprio boom di interesse con articoli di riviste, siti Web, corsi di istruzione, conferenze e molti libri.

In effetti Java presenta alcuni vantaggi rispetto ad altri linguaggi come C++ e Visual Basic: in particolare è piccolo, semplice, sicuro, orientato agli oggetti, indipendente dalla piattaforma, multithreading.

Java è piccolo e semplice

Java è stato descritto come un "C++ meno", perché è nato dal C++ con l'omissione di alcuni elementi. Chi conosce il C++ o il C non ha difficoltà ad apprendere l'utilizzo di Java. Molti elementi sono stati rimossi rispetto al C++ per ragioni di semplicità: si tratta di elementi difficili da utilizzare e perciò fonti di potenziali errori, come i puntatori e la gestione della memoria. Trovare un errore di puntatore in un programma di una certa dimensione è quasi un'esperienza mistica. La gestione della memoria in Java viene eseguita automaticamente, perciò i programmatori non devono preoccuparsi di scrivere routine di garbage collection.

Un'altra scelta che ha contribuito alla semplicità del linguaggio è rappresentata dall'adozione di tipi e oggetti elementari. Il linguaggio insiste molto sulle regole per le variabili (quasi sempre occorre utilizzare le variabili con il tipo di dati dichiarato, o eseguire casting espliciti per manipolarle).

In questo modo è possibile rilevare un errato utilizzo di una variabile al momento della compilazione, invece di dover attendere il programma eseguibile per osservare il problema. Il risultato è che il comportamento del programma è più facile da prevedere.

Java è orientato agli oggetti

La programmazione orientata agli oggetti è un modo potente di organizzare e sviluppare il software. In breve, la programmazione orientata agli oggetti organizza un programma come una serie di componenti chiamati *oggetti*, che sono indipendenti l'uno dall'altro ma seguono delle regole per comunicare tra loro.

Java eredita i concetti orientati agli oggetti dal C++ e da altri linguaggi come Smalltalk. Il fatto che un linguaggio di programmazione sia orientato agli oggetti potrebbe risultare sgradito a qualcuno, poiché la programmazione orientata agli oggetti impone una sorta di timore reverenziale, almeno all'inizio, anche a chi ha qualche esperienza di programmazione in altri linguaggi. Tuttavia, i programmi orientati agli oggetti sono più adattabili per l'uso in progetti diversi, più facili da capire e meno inclini agli errori.

Il linguaggio Java comprende una serie di librerie di classe che mettono a disposizione tipi di default per le variabili, capacità di input e output e altre funzioni. Contiene anche classi per il supporto della rete, protocolli Internet e funzionalità dell'interfaccia utente grafica.

La comunità dei programmatori è in fermento, perché Java offre una nuova opportunità di utilizzare le tecniche orientate agli oggetti in campo pratico. Smalltak, il linguaggio pioniere della programmazione orientata agli oggetti negli anni '70, è apprezzato ma non si è mai diffuso come linguaggio per lo sviluppo di applicazioni concrete. Il C++ è orientato agli oggetti, ma i concetti principali del suo utilizzo sono stati già descritti. Java sta superando tutte le barriere per quanto riguarda la popolarità, soprattutto nel campo della programmazione su Internet e dello sviluppo di applicazioni distribuite.

Tim Berners-Lee. L'inventore del World Wide Web, ha spiegato il motivo per cui è interessato a questo linguaggio: "Ora abbiamo una scusa per utilizzare davvero la programmazione orientata agli oggetti".

Java è sicuro

Un altro aspetto fondamentale di Java è la sicurezza, che ha rappresentato uno dei motivi del successo del linguaggio. Un programma Java eseguito da una pagina Web è detto *applet*, mentre gli altri programmi Java sono chiamati *applicazioni*.

Quando viene riscontrato un applet in una pagina Web (se l'utente dispone di un browser compatibile con Java), il browser lo preleva insieme al testo e alle immagini della pagina, quindi l'applet viene eseguito sul computer dell'utente. Tutto ciò può risultare pericoloso, perché l'esecuzione di un programma può portare a virus e altri problemi.

Java fornisce diversi livelli di sicurezza. Per prima cosa, il programma è stato progettato in modo che risulti estremamente difficile eseguire codice dannoso. L'eliminazione dei puntatori rappresenta un notevole passo in avanti, da questo punto di vista. I puntatori sono una caratteristica potente, come possono testimoniare i programmatori in C, ma possono essere utilizzati per accedere a parti del programma che dovrebbero rimanere nascoste e alterarle. Con l'eliminazione di tutti i puntatori, fatta eccezione per una forma limitata di riferimenti agli oggetti, Java risulta un linguaggio molto più sicuro.

Un altro livello di sicurezza è il verificatore di bytecode. Come si è detto in precedenza, i programmi Java sono compilati in una serie di istruzioni chiamata bytecode; prima dell'esecuzione, un verificatore controlla ogni bytecode per garantirne l'integrità.

Oltre a queste misure generiche, ve ne sono molte altre che riguardano specificatamente gli applet. Per impedire a un programma di danneggiare l'unità a disco del computer, secondo l'impostazione di default gli applet non possono aprire, leggere o scrivere file sul sistema dell'utente. Inoltre, poiché gli applet possono aprire nuove finestre, queste sono identificate da un logo di Java. In questo modo si evita che qualcuno possa falsificare una finestra di dialogo per l'inserimento della password.

L'ultima versione di Java (1.1) offre un approccio molto più avanzato alla sicurezza, che consente di utilizzare firme digitali a scopo di verifica. In altre parole, è possibile creare un applet e apporvi una sorta di timbro identificativo, che gli utenti devono essere in grado di verificare. In questo modo si perdono le restrizioni delle vecchie versioni di Java, ma si assegna una responsabilità ancora maggiore all'utente.

È un po' come lo spioncino della porta, che consente di dare un'occhiata fuori per decidere se lasciare entrare una persona o meno.

Tutte queste norme di sicurezza non garantiscono la certezza di essere al riparo da occhi indiscreti. Diversi esperti di sicurezza hanno trovato vari modi per aggirare le protezioni degli applet, già durante il primo anno di vita di Java, e i dettagli sono stati inviati a JavaSoft e agli sviluppatori di browser. In futuro saranno certamente scoperte altre falle nel sistema di sicurezza di Java. Per non correre rischi si possono utilizzare gli applet disponibili presso il sito di Gamelan (`http://www.gamelan.com`*), che sono stati verificati e certificati da un'autorità esterna.*

Java è indipendente dalla piattaforma

L'indipendenza dalla piattaforma è un altro modo di dire che Java è neutro rispetto all'architettura del sistema. In pratica significa che per i programmi Java non conta il tipo di sistema su cui vengono eseguiti.

La maggior parte del software è sviluppata per un sistema operativo specifico. L'*indipendenza dalla piattaforma* è la capacità di un programma di lavorare su vari sistemi diversi. Java è del tutto indipendente dalla piattaforma.

I tipi di variabili di Java hanno la stessa dimensione in tutte le piattaforme di sviluppo di Java, perciò un intero è sempre della stessa dimensione, indipendentemente dal sistema utilizzato per la compilazione. Inoltre, come è testimoniato dall'uso di applet nel Web, un file di Java `.class` di istruzioni in bytecode può essere eseguito su qualsiasi piattaforma senza alterazioni.

Esistono interpreti Java in grado di eseguire programmi per Microsoft Windows 95 e NT, Apple Macintosh 7.5, SPARC Solaris 2.3 e Intel x86 Solaris. Per altri sistemi sono in fase di sviluppo nuove versioni dell'interprete.

Java è anche altro

Finora non si è menzionato il multithreading. I *thread* rappresentano un modo con cui un programma è in grado di eseguire diversi compiti contemporaneamente. Molti sistemi operativi sono multitasking; Windows 95, ad esempio, consente di scrivere un documento in Word mentre si utilizza Netscape Navigator per prelevare un'immagine da Internet.

Un linguaggio multithreading estende questo comportamento ai programmi, che consentono dunque di eseguire diverse serie di istruzioni in modo concorrente. Java mette a disposizione gli strumenti per scrivere programmi multithreading e fare in modo che tali programmi siano eseguiti in modo affidabile.

Un altro aspetto di Java è la sua natura incentrata sulla rete. In questo campo Java si distingue nettamente da tutti gli altri linguaggi di programmazione. Nelle parole di uno dei creatori del linguaggio, James Gosling, "L'aspetto che distingue Java è il suo approccio alla programmazione distribuita.

JavaOS

Nel maggio del 1996 JavaSoft ha annunciato di lavorare allo sviluppo di JavaOS, un sistema operativo compatto destinato all'esecuzione di programmi Java. Lo scopo ufficiale era quello di creare la più piccola piattaforma che potesse supportare Java.

Al momento JavaOS è disponibile per varie architetture di sistema. Un'implementazione di rete completa, che comprenda il browser HotJava, le librerie di classe e otto megabyte di font, richiede 4 MB di spazio su disco e 4 MB di memoria libera. JavaOS e HotJava insieme utilizzano meno di 2,5 MB di RAM, lasciando oltre 1,5 MB per la gestione delle pagine Web nella casche, di dati o di immagini e di applicazioni.

Il domani

Il prossimo anno sarà un banco di prova: se Java vuole rimanere il linguaggio di elezione per la programmazione orientata agli oggetti, deve soddisfare alcune richieste. La presentazione della versione 1.1 ha zittito molti scettici, ma finché non vi sarà qualche prodotto commerciale scritto interamente in Java, continuerà a esistere lo scetticismo.

Struttura del libro

Questo libro è suddiviso in nove parti dedicate ad approfondire aspetti particolari del linguaggio e descritte di seguito.

Parte 1

Ogni esploratore deve partire da qualche luogo, e qui si parte per il grande mondo di Java. Viene spiegata la natura di linguaggio di programmazione orientato agli oggetti di Java e il suo senso pratico.

Parte 2

Lo scoglio più difficile da affrontare per l'utilizzo di Java è il linguaggio di programmazione. Non basta conoscere il linguaggio in termini generali, occorre anche essere a conoscenza dei dettagli interni delle classi e delle loro relazioni con gli altri aspetti di Java. Inoltre vengono spiegati i termini *thread* ed *eccezione*, che a molti fanno tanta paura.

Parte 3

I package rappresentano la più potente caratteristica di Java: essi mettono a disposizione funzioni matematiche, di input/output, di rete e di sicurezza. In questa parte viene inoltre spiegato come gestire l'interfaccia utente con i package standard del linguaggio.

Parte 4

Java ha numerosi punti di forza, ma gli aspetti che saltano immediatamente all'occhio dell'utente e dello sviluppatore sono l'interfaccia utente e la grafica. L'AWT gestisce tutte le caratteristiche visuali di Java, fornendo il supporto per i componenti dell'interfaccia utente e per l'elaborazione delle immagini a livello avanzato. Nella versione 1.1 sono stati introdotti molti miglioramenti in questo campo.

Parte 5

Un linguaggio di programmazione per Internet deve necessariamente essere ricco di funzionalità di rete, descritte in questa parte, dove viene inoltre spiegato come scrivere un applet multiutente in Java.

Parte 6

Con questa parte si entra nelle aree più avanzate della programmazione in Java. In particolare vengono trattati il debugging, la persistenza e la serializzazione, tecniche utili per memorizzare e ripristinare lo stato di oggetti Java.

Parte 7

L'idea di raccogliere tutti i file e le risorse richiesti per un applet o un'applicazione in un unico archivio compresso e protetto è fondamentale per la sicurezza e l'efficienza nella distribuzione dei prodotti Java.

Parte 8

La nuova versione 1.1 di Java offre ampie funzionalità per il supporto dei database, che mancavano nelle versioni precedenti. Java supporta i database con l'API JDBC, che offre la possibilità di interagire tramite SQL con tutti i tipi principali di database.

Parte 9

Nell'ultima parte si trovano le appendici con la sintassi del linguaggio, i diagrammi della struttura di classi e altri strumenti e informazioni utili.

PARTE 1

Nozioni di base

IN QUESTA PARTE